

ALPHA

MICRO

USERS

SOCIETY

Newsletter

VOL 2 NO 7 AUG, SEPT, 1979

ALPHA MICRO USERS SOCIETY
1216 Pearl St., P.O. Box 1723, Boulder, Colo. 80306
(303) 449-6917

BOARD OF DIRECTORS

OFFICERS

MEMBERS

President
James Taylor, DPCI, Inc.
Boulder, CO

Vice President
Eugene Platt, North American
Title Co., Houston, TX

Second Vice President
Burl Hubbard
Lubbock, TX

Secretary Treasurer
Dick Leach
Applied Computer Systems
Golden, CO

Stan Viet
New York, NY

Guruprem Khalsa
Khalsa Computer Systems
Pasadena, CA

John Lindsey
David A. Bell, Inc.
Los Angeles, CA

Ken Biven
KB Communications
Seattle, WA

John Zielinski
Arizona Computer Systems
Jerome, AZ

STAFF

Secretary - Sharon Greene

Newsletter Editor - Steve Elliott

AMUS office hours are from 8:00 AM to 5:00 PM, Mountain time. Our overworked secretary is Sharon Greene who is happy to assist you with any question you might have about AMUS, the location of a piece of application software, getting help with a particularly nasty bug, or whatever. We're here to help.

AMUS has a library of programs that have been donated by members for distribution to other members. Programs are available either through the AMUS network, or via floppy discs. Discs may be ordered through Sharon Greene at a cost of \$10.00 each.

THE AMUS NEWSLETTER is available for \$25.00 per year which includes membership dues in AMUS. See the back inside cover for membership form. Additional copies of the newsletter are available to members at \$7.50 per year. The newsletter welcomes any bugs, fixes, articles, reviews of software, opinions, advertisements, or program listings concerning the Alpha Micro computer. Dealers are asked not to submit advertisements for products available through other local Alpha Micro dealers. Hardware manufacturers and software authors are encouraged to provide AMUS with descriptions of their product for inclusion in the newsletter. Please submit material as high contrast, camera ready copy on 8 1/2 by 11 inch white paper.

TABLE OF CONTENTS

| | |
|--------------------------------------------------|----|
| From the President..... | 1 |
| AMUS computer network..... | 5 |
| Letters from members | |
| Dr. James Phelan: help with marathon..... | 6 |
| Dr. James Phelan: review of Reno programs..... | 8 |
| Khalsa bug list for 4.2..... | 11 |
| Khalsa notes on 4.2..... | 15 |
| Khalsa monitor size information..... | 17 |
| Phil Putnam: needs system for legal office..... | 20 |
| Eastern College: grant award announcement..... | 21 |
| Steven Orr: auto number program..... | 22 |
| Jeff Oberholtzer: modification of FREBLK.MAC.... | 24 |
| Mary Sue Lee: needs help with Calcomp T80 disk.. | 27 |
| Payne Jackson & Assoc.: Trident backup system... | 28 |
| Current software list..... | 32 |
| Software announcements | |
| Data Masters: accounts receivable..... | 36 |
| D.P.C.I.: insurance system..... | 37 |
| Oberholtzer Engineering: Alpha Word..... | 39 |
| Real Oregon Computer Co.: utility packages..... | 40 |
| Byte Shop of Reno: Floating point conversion.... | 43 |
| Membership application form for AMUS..... | 48 |

FROM THE PRESIDENT

Eugene Platt in Houston is interested in co-oping with someone to study the COBOL compiler from CAR in England. Any takers should call (713) 681-6949.

Len Bacus has informed us that Alpha Micro is negotiating with CAR for their COBOL compiler. If you are interested in helping to test and analyze the package please let Len and AMUS know.

A number of people from educational institutions have contacted me about wanting to talk to other educational users. If all of you who are interested would call or write to Sharon at (303) 449-6917 we will coordinate an educational users group and put you in touch with one another.

Rumor has it that Alpha Micro's FORTRAN project is not going well.

Elsewhere in the newsletter you will find information about the Communications Network. Many thanks to Alpha Micro for providing the boards to AMUS and to Dick Leach for offering to provide the rest of the equipment at cost.

When using TRM:TERM to print from any other terminal to a printer, the TAB (-1,X) function will not work. The system checks to see if you are attached to the terminal.

COMPIL and BASIC seem to have some unknown differences. We have a program that will work if compiled under BASIC but not if compiled under COMPIL.

IMAGE MODE - DELETE

The system crashes easily when a number of characters are deleted with the DEL or RUB key if you are in image mode. The problem is one of timing and may be alleviated by deleting very slowly. The bug has been found and corrected in the terminal service routine but did not get into the 4.2 release.

SLEEP

SLEEP (return) tends to sleep forever. Since the documentation says to type in the number of seconds to sleep, the program was changed from clock ticks to seconds.

DSKCPY

Careful - DSKCPY will accept nulls as input for the drive prompts and set them to zero.

HP2621 A/P

AM software has received some requests for a driver for the HP2621 A/P terminal. They think it is too expensive for what it does, but if someone would like to loan them a terminal...

DEALERLESS USERS

Any A/M users who are now without dealer representation can and should get help. Since Alpha Micro does not deal directly with end users, you may find yourself currently or in the future in a bind if you need help with something.

You should write Alpha Micro Marketing and get them to assign you a dealer who can service you. Please send a copy of your letter to AMUS and we will follow up and make sure something is done.

CURRENT AVAILABILITY QUOTES

| | | |
|-------------|-----------------------------------------------------------------------------------------------------------------------|-----------------|
| AM-210 | Dual-density, dual-sided floppy disk controller supporting CDC, REMEX | Sept. 79 |
| AM-310 | Serial, buffered I/O communications board. | Sept. 79 |
| AM-410 | CDC Phoenix 90 MByte drive and controller board. | Available now |
| AM-600 | Tape transport and controller. No reliable tape transport has been decided on. | ? maybe Oct. 79 |
| AM-1011 | Sixteen bit processor, double high boards, cabinet or rack mounting in an A/M designed cabinet with 2.4 MByte floppy. | Mid Oct. 79 |
| AM-1031 | Same as AM-1011 but with HAWK 10MByte drive. | Late Oct. 79 |
| AM-1051 | Same as AM-1011 with the Phoenix 90MByte drive. | Late Oct. 79 |
| 5.0 Release | New file handling techniques, new security system, synchronous I/O. | June 80 |

PERSCI

The people at persci are now supposedly quoting 30 day turnaround for maintenance. However, my drive had already been there for seven weeks when I talked to them and it took two more weeks before it was shipped out. They were going to send a letter describing this new philosophy and capability but we have not received it yet.

Persci says that the model 277 is upgradable to a dual density drive. Alpha Micro says that it won't work. The model 299 is a dual sided, dual density drive. One was loaned to Alpha Micro for testing with the new AM-210 board but Persci took it back before it was working so Alpha Micro doesn't know what the status is. The 299 has undergone a complete revision and version B

supposedly works, but we still don't know if it will work with the AM-210. Persci is also offering a stripped down version of the 299 known as the 288.

WANCO

Wanco sold the floppy drive division to someone who we believe has now resold it to Siemars. It is our understanding that they will discontinue the drive. Alpha Micro has sold their supply of Wancos but will continue to support the drives now in use. The single density Wanco is upgradable to dual density and it does work on the AM-210 controller.

CALCOMP-TRIDENT

The Calcomp hard disk division has been sold to Xerox. At this point we assume that the Trident series is alive and well just with a different label. Calcomp's floppy disk division was also sold, but to Billings computer systems in Utah.

SOROC

It has been reported that turning off the SOROC will sometimes create errors on the Hawk disk that you cannot correct even with DSKANA or DSKCPY. Alpha Micro apparently has run into the same problem and has no solution. Obviously there is a reason and I'll bet someone knows what it is and is hiding it from us...come on, out with it.

HAZELTINE DRIVER 4.2

We might as well face it...they'll never get it right. After the XTBL: label the instructions should look like this:

```
BYTE 137,177 ; underscore becomes delete
BYTE 177,137 ; delete becomes underscore
BYTE 16,36 ; control N to home
BYTE 0
```

The only thing wrong with HAZEL.TDV was that the control N didn't work in VUE.

ALPHA ACCOUNTING RELEASE 1.1.1

Everyone that is licensed for the accounting package should have received release 1.1.1 by now. Rather than being a complete release on a Hawk pack it is simply a document showing the corrections to the bugs on 1.1. Unfortunately for those of us who are trying to get Order Entry running, the release only points out the bugs, it doesn't fix them.

Also, there are several errors in 1.1.1:

On page 2 the corrections to the accounts receivable expand program should be made to ARXPND not to APXPND.

On page 4 under program SLSJNL you will find a correction to line 2250. It should be line 2550.

On page 7 under GLTRX, line 1360 did read...RECCNT + 1..
not...RECCNT + 2 as the correction indicates.
The correct instruction is...RECCNT + 1...

Also on Page 7 (thanks to Sue in Applications for this
one) the correction for line 1670 is wrong.
Line 1670 should be left as is and the following
line should be added:
1645 PFC = ENTRY [1,3] using "#ZZ"
Also, line 1940 did read ENTRY = "0"

On page 9 under SOURCE, line 1090 did read...[1,1])...
it should be...[1,3])...

On page 12 under UPDINV, line 1570 has the same problem
as 1270 i.e. part of the variable name is in
lower case. It won't hurt anything though, because
the variable is not referenced in the program.

Jim Taylor

AMUS COMPUTER NETWORK

As you know, we have been predicting an AMUS computer network for some time now. The last feeble attempt planned to make use of a local computer that was connected to TELENET and things progressed up to the point where we started to put programs onto the computer and found that their charges for storage were by the day and not by the month as we believed. This minor budgetary consideration left us with no network, and back at the drawing board. Since then some new developments have brightened up the network picture.

Alpha Micro has donated an AM-100, AM-300, AM-500 and 64K of memory to AMUS. It remains the property of Alpha Micro and goes back to Irvine if we should disappear, but it is ours to use as we wish, and Alpha Micro will cover the maintenance of the four boards

The J.F. Hurlbut Co., a local Alpha Micro dealer, has offered to sell the equipment needed to round out a 10MByte Hawk system able to communicate over phone lines to AMUS at cost. This would come to about \$7,500 for terminals, the power supply and box, Hawk disk, and phone communication equipment.

AMUS now has roughly \$8,000 in savings and \$2,500 in its checking account.

According to our by-laws, we need to have the approval of the membership before spending this amount of money. All AMUS members are asked to get in touch with the AMUS office and voice their opinions concerning the allocation of funds for creating a computer network. You may call or write, whichever you prefer, but please be sure we hear from you before October 10, 1979.

Some things you may wish to discuss are:

1. Do we want to spend the society's money on a computer, or is there an alternative we should be considering.
2. Do we wish to make this available only to AMUS members, or open to all.
3. Do we wish to charge for use of the network. If so, do we charge one rate to members, and another to non-members. (If they are allowed on the system).
4. What considerations should be made for preparing software for distribution. Especially, what documentation should be required, should there be a software editor (or some other entity) that kicks the tires of any program submitted before it is allowed to be placed in the general distribution account.
5. What services would you want.
6. What services would you be able to help with.

Ed note: From time to time Dr. Phelan sends us great letters, unfortunately he doesn't ever change the ribbon in his printer so I have to type them in by hand.

July 22, 1979

James H. Phelan, M.D. 18953 Memroial Drive North Suite O Humble,
Texas 77338 713/446/8119

Print: Need a way to kill whole print queue. PRINT *.*/*K won't work after the first file has been printed. Will get "*****.*** not in print queue". KILL SPOOL doesn't seem to work either.

Bayou City Fun Run:

Am looking for ways to interface my AM-100 to device(s) that will help with the following:

1. Keep track of INDIVIDUALS crossing finish line and queued in finish chutes.
2. Keep track of TIMES of finishers corssing the finish line.
3. Be able to handle 10,000 runners.
4. Be able to be used in field where power supplies, telephones, weather, etc. may not be optimal, i.e.: cassette recorders, battery power supply, bubble memory, etc.
5. Means of identifying runners must be waterproof (sweat, rain, Gatoraid) and mutilation proof.

6. Must require minimal participation of runners or officials to avoid human error under stressful conditions.
7. Reasonable cost (under \$10,000 for 4 finish lines)
8. Require as little clerical effort as possible.

We are considering:

1. Adapting our Chronomex clock for digital output to: cassette, bubble, paper tape. Modem - cassette - modem - computer.
2. Bar code readers
3. Employee badge readers
4. Magnetic card readers.

We would appreciate any hardware or software ideas.

MUMPS:

Many people would like to see MUMPS for the Alpha Micro. Anything on the horizon yet?

THE ALPHA MICRO SYSTEM:

Enclosed is part of a letter from an associate re: the A/M. Being fairly ignorant of the nuts and bolts of the system, would anyone here like to answer his questions?

The AMS 100 is great, reasonably priced, but suffers from lack of utilizing the vast 8080 software such as that available from the CP/M user's group. The BASIC seems rather good and incorporates a lot of what are "system" software commands in other installations. At the same time, commercial software for this system is available, but priced beyond the pale for an individual user. Your CDC Hawk is enviable, I will certainly go that way soon, but demand compatability with the CP/M floppy disk system to support my many correspondents (see the new ICOM hard disc system 4511). Please tell me: Why Alpha Micro?

For example: I recently had a bonified head-crash and lost an entire disc that contained all the data for the SMAC 24-channel autoanalyzer results of some 256 industrial normal patients. This was on the Altair system, and through the use of my DISC-FIX I was able to restore all of the data. In a related matter, owing to pure stupidity, I erased an entire disc containing the results of some 56

dogs that we were using for a normal value study. Delving into the CP/M system, I discovered that I had not really erased all of the data, but only altered a code sequence that marked those files available for overwriting. Utilizing my "Dump", I was able to rewrite those sections on the disc and recover my data completely. Is this readily available on the AMS-100?

My prejudice is that AlphaBasic can run rings around any other BASIC and many other high level languages as well. Its multi-user, multi-tasking ability is fairly unique in its class. I don't know the other systems he's talking about well enough but suspect that DSKANA, DSKDDT, DSKDMP, DSKFIL, DUMP, etc. would do what he's asking. The fact that AMOS makes automatic .BAK files whenever a file is edited would aid his problems. I have been impressed by the continual improvement of the AMOS releases and by the aptient support of Alpha Computer Solutions with my dumb questions.

James H. Phelan, M.D.

Ed note: The CP/M system is a fine, inexpensive system, but suffers from its inability to utilize the superior capabilities of the Alpha Micro operating system and AlphaBasic features such as unlimited variable name length and mapping of file records. Fortunately the business applications are sophisticated enough (and priced accordingly) to discourage amateurs. Sequential file mishaps can usually be repaired through the use of the DSKFIX program created by Khals& Computers and available from the users society. Accidentally erased random access data files can also be recovered since the ERASE command only deletes the file name from the directory. Recovering random files consists of entering BASIC and allocating those files again (using the correct number of blocks of course). beginning with the file that was last on the disk and working backward.

Ed note: James also send some good info to us back in June, but we didn't have any room to put it into the newsletter then. These are his comments on the programs offered by the Byte Shop of Reno. If you have purchased software and have comments about it you would like to share with members, we would be happy to have them.

NAME James H. Phelan, M.D. DATE June 1979
 COMPANY physician's office
 ADDRESS (office) 18953 Memorial Drive North Suite C Humble, Texas 77339
 SYSTEM CONFIGURATION:
 AMOS Version 4.1

BYTE SHOP OF RENO, AGAIN:

CLINE.SBR: Works after making corrections as instructed, but second example is wrong; needs extra argument: XCALL CLINE, AS,C

| | |
|------------------------|--------------------------|
| 10 ! TEST.BAS, EXAMPLE | 10 ! TEST.BAS, CORRECTED |
| 20 MAP1 Z\$,S,80 | 20 MAP1 Z\$,S,80 |
| 30 INPUT A,B,C | 30 INPUT A,B,C |
| 40 XCALL CLINE,Z\$ | 40 XCALL CLINE,Z\$,C |
| 50 PRINT A,B,C,Z\$ | 50 PRINT A,B,C,Z\$ |
| 60 END | 60 END |

| | |
|---------------------------------|----------------------------------|
| RUN TEST | RUN TEST |
| ? 1,2,3 | ? 1,2,3 |
| 1 2 3 ,2,3 | 1 2 3 1,2,3 |

RENJ4: Works
 RESEQ: Works
 ABBREV: Works
 ALLOC: Works
 AUTO: Works
 CMPALL: Works
 DATCNV: Seems to work, minor bug fixed per their instructions, never noticed it. Some unspecified things are happening but it's probably my program. New version selling for \$25 looks good on paper. Wish they'd donate it like Khalsa did.
 DING: Works
 GETCH: Works
 IF: Works
 PRINT: Works, but use PRINT.PRG 4.1 instead
 We need a PRINT switch that will allow operator to change paper at /PAGE boundaries so I can print a > 1 pg letter on my letterhead which is NOT continuous fan-fold.
 BASIC: TAB does funny things sometimes, especially when TAB(-1,11) (protect) is on.
 VUE: Please iron out BUSS ERROR's !!

Consulting Editors: Would be willing to serve as consulting editor / program evaluator for medical programs. Have been looking over several office systems but havn't seen a really good one yet. Am presently working on SHOTS to organize and interpret immunization records.

Enclosed is next generation MENU.BAS using GETCH.SBR from BYTE SHOP of RENO. It forces fewer choices per page and provides lazy choice entry without having to enter RETURN all the time:

```
!
! MENU.BAS A MULTI-PURPOSE MENU PROGRAM
! USES GETCH.SBR from Byte Shop of Reno
! Maximum of 9 items per menu page (but may branch to another page)
ON ERROR GOTO END
STRSIZ 90
DIM TITLES$(25),PROGRAM$(25)
INPUT "",FILENAME$ !INPUT DATA FILE NAME FROM *.CMD FILE
START:
PRINT TAB(-1,0); !CLEAR SCREEN
OPEN #1, FILENAME$, INPUT
INPUT LINE #1, TITLES$
PRINT TITLES$
PRINT
I = 1
PRINT " 0 "; "-- RETURN TO MAIN MENU --"
LOOP:
INPUT #1, TITLES$(I),PROGRAM$(I)
IF TITLES$(I)="" GOTO CHOOSE
PRINT I;TITLES$(I)
I=I+1
IF I = 11 PRINT CHR(7); "TOO MANY CHOICES!" : GOTO END
GOTO LOOP
CHOOSE:
CLOSE #1
I = I - 1
PRINT
PRINT "What is your pleasure?..."
PRINT"TOUCH line # of desired program: ";
XCALL GETCH,A$
PRINT
IF A$ = "@" GOTO EXIT
IF A$ = "0" CHAIN "DSKG:M.CMD[2,2]"
LINENUMBER=VAL(A$)
IF LINENUMBER = 0 PRINT CHR(7) : GOTO START
IF LINENUMBER > I PRINT CHR(7) : GOTO START
CHAIN PROGRAMS(LINENUMBER)
END:
PRINT CHR(7)
EXIT:
```

Once again, Bob Fowler of Khalsa Computers has provided us with an extensive list of the bugs he has discovered in AMOS version 4.2. Bob obviously spends considerable time and effort on this project, and he deserves a round of applause for doing a fine job.

AMOS BUGS (Version 4.2)

8/27/79

(01) FMT200 --- Crashes system if disk head is loaded

On a Persci-AMS system, put a scratch diskette into drive 1, and type
LOAD FMT200
(eject system disk to protect it)
FMT200 DSK1:

If the disk head was loaded at the time that the FMT200 command was typed, then AMOS crashes, no formatting occurs, and the disk head retracts after the normal HEDLOD time. If the disk head was NOT loaded at the beginning of the FMT200 command, then the format will be successful, but the disk head will NOT retract after the normal HEDLOD time.

I tried this on 2 independent sets of hardware, with same results. My suggestion is to use the old FORMAT,AMSFMT,WAMFMT,WNGFMT programs from 4.1 until version 4.3 comes out. This will avoid crashing the system, and possibly crashing both diskettes in the drive (this happened to me).

(02) MONGEN --- fails in some cases

Starting with a Persci-AMS Version 4.2 Update Diskette (staight from Alpha Micro), create the "same" monitor as the one already on the disk, and it will fail to boot the system. To be specific, use FIXDVR to make an AM200/Persci/AMS driver, then use MONGEN to imbed it within the original monitor, but give it a new name (NEW.MON is good), SAVE it onto disk, then use MONTST NEW.MON to test it. System will not boot. Copy NEW.MON into SYSTEM.MON, and it will still fail to boot.

Also, funny things happen to the hash code on this new monitor when it moves around --- try LOADING any program into memory before using MONGEN (as described above), then create NEW.MON, check its hash code with MAP, DElete the other program from memory, and check NEW.MON's hash code again (it is changed).

(03) Disk Drivers Documentation --- Trident misprint

In AMOS document DWM-00100-32, page 2, section 2.0.4, line 4, the first "T-80" should actually read "T-200". This misprint is confusing because a lot of similiar numbers appear in this section.

(04) LPTINI --- log in requirement clarified

If you do not LOG in your Spooling job before FORCEing it to run LPTINI, then LPTINI will give no error message (even if you ATTACHED the spool job to a real crt for spooler de-bugging purposes). This has caused a slight confusion to at least one user. Also, if you LOG the Spooling job into a PPN other than [1,2] before running LPTINI, then LPTINI will re-log it into [1,2] before bringing up the spooler. This auto-log feature is an improvement over previous releases; in these releases, it was always possible to find out what the PPN [1,2] password was by simply listing the SYSTEM.INI file, and looking for the line that immediately followed the "LOG 1,2" line in the spooler section.

(05) printr.INI --- DEFAULT doesn't work

The DEFAULT line in a Spooler INI file currently has no effect whatever. Spool requests are still passed off to all of the spoolers on a "who's-got-the-shortest-queue" basis.

(06) DSKPAK --- It works !

Not a bug report. It works. To get the most packing possible, you may have to run DSKPAK more than once.

(07) BASIC --- SCALE interferes with TAB(-1,0)

Enter the following lines to demonstrate this problem :

```
BASIC
10 PRINT TAB(-1,0); "TEST";
20 SCALE 1
   SAVE FRED
   BYE
   BASIC
   LOAD FRED
RUN
```

The screen will not clear. If the 4 indented lines above are omitted from the typing sequence, then the screen WILL clear properly. A temporary fix for now is to insert SCALE 0 before each TAB(-1,0), and then insert SCALE 1 immediately after (or whatever other SCALE factor was in effect).

(08) BASIC --- A\$ USING B\$ has a problem

Please ignore/scratch out bug #8 from buglist #11. The problem was own failure to use a larger STRSIZ than the (default) 10 value, and this caused to be truncated to 10 characters.

-- Bob Fowler

(09) COMPIL --- Bad syntax is slipping by

Using VUE or EDIT. put the following into a new file :

```
10 DIM A(10)
20 B=A(N+)
```

Note the bad syntax in line 20. BASIC will not let line 20 be entered. However, if we use COMPIL on this file, it will let line 20 by with no error messages. A line similiar to this was contained in a laaaaarge BASIC program of mine, and this bad syntax eventually caused error messages at execution time in a totally unrelated part of the program. It was hard to track down

(10) COPY --- Slightly incorrect error message

If FRED.DAT is a random file, then AMOS will respond to this command
 COPY TRM:=FRED.DAT
 With the error message "% Random files cannot be transferred to MEM:",
 rather than " TRM:".

(11) LOG --- Associated Ersatz names not reported

According to the new manual, typing the following line :
 LOG DSKO:[1,4]
 Should cause the following response from AMOS :
 Logged into DSKO:[1,4]
 Ersatz name is SYS:
 The second line of the response does not, in fact, appear.

(12) VUE.PRG --- UNYANKING too many lines inserts a blank line in file

If VUE currently has m lines of text in its memory, and you execute an
 UNYANK n
 command (where n is any integer larger than m), then a blank line will be
 inserted immediately following the last line that was in VUE's memory.
 This happens both with small files & files larger than VUE's memory size.

(13) ADM3.TDV -- TAB(-1,n) problems

- (a) TAB(-1,5) and TAB(-1,6) are reversed ! This is easy to fix, but it is surprising that no one noticed this before ! The comments in the ADM3.MAC are correct, but the codes sent out are reversed.
- (b) TAB(-1,9) leaves cursor at end of line. This is the "clear to end of line" code, and it is simulated in ADM3.TDV by sending out enough blanks to over-write the rest of the line. Unfortunately, ADM3.TDV does not restore the cursor to its original position afterwards.
- (c) TAB(-1,10) is not implemented (clear to end of screen), but could be.

See the latest AMOS newsletter (1979 Jun/Jul, page 27-31), for a listing of the modified ADM3.MAC by C. Brandin of Business Operating Systems in Colorado. It corrects the bugs (a),(b) and implements (c).

(14) ADDS.TDV --- TAB(-1,n) still disagree with standard codes

The 4.2 supplement to the ALPHABASIC manual gives new TAB(-1,n) codes for n=17 to 27. These agree with all the terminal drivers, except the ADDS.TDV driver. whose codes for n=17 to 21 should be changed.

(15) BASORT.SBR --- Floating variables must be on even (word) boundaries

A note in the 1979 Jun/Jul AMUS newsletter from Business Information Systems says that if BASORT doesn't sort a particular floating variable properly, then "add another byte in the MAP ahead of the variable", and it will sort properly. Indeed, BASORT appears to require that all F6 variables passed to it lie on even byte locations. This can be accomplished most easily by putting all such F6 variables into MAP1 statements. MAP1 variables are always placed on even byte positions. If necessary, a "filler" byte is left by BASIC to do this. All other MAP levels are forced into whatever odd/even position is next available. If you must put an F6 variable on a MAP2 (or lower) level, then you will have to keep track of the odd/even positions yourself and, if necessary, insert 1-byte "filler" variables to accomplish what BASIC does in MAP1s. This is exactly what is done in the Alpha Accounting System to get around this problem.

It would be a good idea to put some kind of note about all this in the BASORT documentation.

(16) SERCH.SBR --- Multiple keys must be consecutive

I am passing this bug on second hand (haven't checked it out myself). If more than one search key is passed to SERCH.SBR, then the search will not be performed correctly unless the 2 or 3 keys are consecutive in the file record. In other words, using the variable names from Alpha Micro's SERCH.SBR documentation,

STPOS2 (if used) must equal ENPOS1+1

STPOS3 (if used) must equal ENPOS2+1

(1) DIR --- add size (in bytes) option

This number is always given by MAP output, and by DIR MEM: output. Allow the user to enter DIR /S if he wants the size (in bytes) of files.

(2) SYSTEM --- (Hawk System) Always indicate which drive is DSKO

As of now, typing SYSTEM on a Hawk system will either respond with a message that the system is running off of the Cartridge Disk, or will omit this message (meaning that the system is running off of the fixed Cartridge. During such procedures as backups, it may be better to include a message in either case; in other words, if running off of the fixed disk, say so. Also, it may be helpful to put this information into the SYSTAT output in an unobtrusive way (e.g., immediately to the right of the "DSKO nnn FREE BLOCKS" line at the bottom).

(3) BASIC --- different continuation line syntax

I am an old Fortran user, and in that language continuation lines are indicated by a character at the beginning of the continuation line, rather than a character at the end of the continued line. Confused? Well, the major advantage of the Fortran syntax is that if you decide to delete the continuation line later, then you need only delete that (second) line. In ALPHABASIC, you would have to delete this (second) line and also change the (first) line, which is irritating.

(4) VUE versus HAZEL.TDV --- The problem with keypad-based TDV's

There are many crt's, and just about as many different sets of cursor control codes (see the section following for more on this chaos). However, all crts do agree on the controls generated for the letters of the alphabet, namely, control-A through control-Z are always sent off as Ascii codes 1 through 26. As long as VUE, DDT, or any other character-oriented program systems stick with these standard codes, life is easy. Implementation of the HAZELtime 1510/1520 keypad controls (up,down,left, right cursor) into the HAZEL.TDV terminal driver has created a driver that is more properly called a HAZEL-1510-VUE driver --- it is inappropriate for use with the HAZELtime 1500, since the 1500 has no control keypad; it is inappropriate for use with other control-code-based programs, because their particular control-codes must be changed to agree with the re-assigned codes in HAZEL.TDV. I speak especially with regard to the Khalsa WORD processor, which uses control codes, and any other applications programs that are (or will be) using control codes.

The heart of the problem is this : any crt (like the HAZELTIME series) which begins any control-code sequences with Ascii codes 1 through 26, will encounter ambiguity problems in a keypad-type terminal driver. Any driver that converts inputted Ascii codes 1 through 26 will be unable to discriminate between whether these codes were entered as alphabetic controls or by keypad cursor-controls (because they generate identical input !)

Unfortunately (see Cursor Control Table) the most persistent ambiguity cases, over all crts, include the upline and forespace controls, and these are among the most commonly-keypadded functions.

(5) CRT Cursor Controls Table --- see enclosed table

Ever want to see, at a glance, the major differences/similarities between CRTs ? Or (from a different perspective) what total chaos can result when there is a total lack of standardization ? See the 1-page (abridged) summary of crt capabilities enclosed. Room for expansion. Comments (and additions !) welcome.

(6) AMOS Monitor Size --- see enclosed 3-page write-up

Dealers : ever get those nagging calls from customers who want to know why their monitors don't fit when things are changed ?
Customers : ever get the feeling that your dealer pulled a "fast" one on you when he said you have 64K, but (somehow) you only got 32K to use ?
Answer : See the enclosed table which gives you the sysgen facts of life, byte-by-byte. Using the figures given, you could even code a BASIC program which would read a given SYSTEM.INI file, and tell you how big the monitor would be, without actually hitting reset !

(7) Persci Alignment --- done at Khalsa

The Jun/Jul AMUS Newsletter mentioned 3 folks who do/might-do Persci alignments. They are in Seattle, Ventura, and Santa Monica. Well, we are in Pasadena. and we do alignments with an average 3-day turnaround. The average price is \$45, done by our own service department, "factory-trained at Persci". If you want to do it yourself, our "tools" are also available for a flat \$50. The tools include a 2708 boot prom and instructions on how to use it. The prom is used in place of the AM200 prom, and loads in a small monitor called ALIGN.MON, which has several alignment-related commands available. The program was written by Larry White (who also wrote the Khalsa Telecommunications Software).

SIZE OF AMOS MONITORS AFTER SYSGEN

This document is meant to aid you in calculating (precisely) the total room necessary to configure a given kind of AMOS system, without having to actually go through the whole sysgen process. It can be used by a salesman & a prospective customer to figure out whether a particular kind of system will actually fit in a prescribed amount of core. It may also be used by the system programmer to "juggle" system size requirements on paper, without having to kick any users off the system to do so.

In general, the AMOS resident monitor consists of a what we shall refer to as the "Base Monitor", whose size (in bytes) is a constant for any given AMOS release. The SYSTEM.MON found in PPN [1,4] consists of the Base Monitor plus an additional area whose size has been exactly 4314 bytes for at least 3 AMOS releases. (Why 4314 ?). This additional area is where the Disk Driver is contained, and where new drivers are inserted by MONGEN. Most drivers are less than 1000 bytes, and the remainder of this 4314 byte area is left unused. On the system disk, this unused area actually takes up disk room (thus resulting in "apparent" SYSTEM.MON sizes of 13000+ bytes), but during sysgen time this unused area is eliminated in core, so that only the Base Monitor and the Disk Driver occupy (precious) memory.

To determine the exact total size of an AMOS monitor, you must first have the following information :

- (1) The AMOS Release Number (e.g. "4.2")
- (2) The Disk Driver used in the Monitor (e.g. "PERAMS")
- (3) A listing of the SYSTEM.INI file (only those lines that precede the line "SYSTEM" effect the size of the AMOS monitor).

The AMOS Release Number is needed because the sizes of various programs change with each release. The Disk Driver must be known, because each driver has a different size. Every single line that precedes the line "SYSTEM" in a SYSTEM.INI file will add to the size of the AMOS monitor at sysgen time; it is possible to insert lines into this part of the SYSTEM.INI file that do not affect the monitor size, but they do not have to precede the "SYSTEM" line (they can be moved to the area following "SYSTEM").

Page 2 contains a summary of the various contributions to the monitor size. The various constants on page 2 are correct for AMOS Version 4.0, 4.1, & 4.2, but there is no guarantee that they will not change in the future. Also, I have not at this time checked any earlier versions.

Page 3 contains the size (in bytes) of various programs for several AMOS releases, with room for future expansion. The user may wish to add a few more programs to this table, based on his own particular needs.

CONTRIBUTIONS TO AMOS MONITOR SIZE

- (-) SYSTEM.MON Size Of Base Monitor
 + Size of Disk Driver
- (1) JOBS For each jobname in a JOBS line :
 + 292 bytes
- (2) TRMDEF For each TRMDEF line :
 + 70 bytes
 + in-width buffer size
 + in-buffer size
 + 2 x (out-buffer size)

 For each different terminal driver :
 + 16 bytes
 + Size of terminal driver (e.g. HAZEL.TDV)
 For each different interface driver :
 + 16 bytes [I only checked this on 4.2]
 + Size of interface driver (e.g. AM300.IDV)

Note : If a TRMDEF line uses the same IDV and TDV as previous lines, then the monitor size is not increased by these drivers a 2nd time. The pseudo drivers PSEUDO and NULL contribute no additional room to the monitor, but any pseudo-terminal still requires 70 bytes + the buffer sizes (see above).

- (3) MEMDEF + 18 bytes for the first MEMDEF line
 + 12 bytes for any additional MEMDEF lines

Note : This data only reflects some simple PIISCEON memory board sample cases; it may require modest revisions.

- (4) DEVTBL + 18 bytes for the first device
 + 8 bytes for any additional devices

Note : If there is no DEVTBL line whatsoever, then (by default) the system still technically has the device "DSK0", but "DSK0" is to be ignored in all our calculations.

Note : Line "DEVTBL DSK1,AMSO,AMS1" has 3 (not 2 or 1) devices.

- (5) BITMAP For each BITMAP line
 + 26 bytes
 + 2 x (size of bitmap in words)
- (6) QUEUE + 16 x (number of additional QUEUE blocks)

Note : This multiplier was 10 in Version 4.0, 16 in 4.1 & 4.2

- (7) SYSTEM prog For each program made resident :
 + 12 bytes
 + Size of program
- (-) SYSTEM [ends all changes to monitor size]

SOME PROGRAM SIZES (in bytes)

| Program Name | 3.2 | 3.3 | 3.4 | 4.0.0 | 4.0.2 | 4.1 | 4.2 |
|----------------------|---------------------------------------------------------|-------|-------|-------|-------|-------|--------|
| SYSTEM.MON | 13122 | 13122 | 13122 | 13256 | 13236 | 13328 | *13364 |
| Base Size | ? | ? | ? | ? | 08922 | 09014 | *09050 |
| Unused | ? | ? | ? | ? | 04314 | 04314 | 04314 |
| Disk Drivers | | | | | | | |
| HWK500.DVR | - | - | 468 | 486 | 486 | 486 | 486 |
| PERAMS.DVR | 472 | 472 | 478 | 478 | 478 | 478 | *(844) |
| PERSTD.DVR | 476 | 506 | 512 | 512 | 512 | 512 | *(844) |
| WNGAMS.DVR | 336 | 340 | 346 | 346 | 346 | 346 | *(844) |
| WNGSTD.DVR | 370 | 374 | 380 | 380 | 380 | 380 | *(844) |
| Other Drivers | | | | | | | |
| AM300.IDV | 342 | 342 | 342 | 342 | 342 | 342 | 342 |
| IMSIO.IDV | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| ADM3.TDV | 268 | 268 | 246 | 246 | 300 | 300 | 300 |
| HAZEL.TDV | 288 | 288 | 294 | 294 | 278 | 314 | 314 |
| HAZEL.KCS | - | - | - | - | 296 | 296 | 296 |
| SOROC.TDV | 288 | 288 | 288 | 288 | 300 | 300 | 300 |
| MEM.DVR | - | - | - | 308 | 308 | 298 | 298 |
| MTM.DVR | 1028 | 1028 | 1028 | 1028 | 1028 | 1028 | 1028 |
| RES.DVR | - | - | - | - | - | 196 | 196 |
| TRM.DVR | 188 | 188 | 252 | 252 | 252 | 252 | 252 |
| AMS,HWK,STD | (look up under original name under Disk Drivers, above) | | | | | | |
| Other Progs. | | | | | | | |
| BASIC.PRG | 11002 | 11056 | 11056 | 11198 | 11198 | 11332 | 11492 |
| RUN.PRG | 9658 | 10440 | 10502 | 10982 | 11000 | 11064 | 11234 |
| DYSTAT.PRG | 902 | 902 | 902 | 902 | 902 | 902 | 902 |
| TODCNV.PRG | 524 | 524 | 524 | 524 | 524 | 524 | 524 |
| EDIT.PRG | 2844 | 2844 | 2844 | 2994 | 2994 | 2994 | 2994 |
| VUE.PRG | - | - | - | 5960 | 8604 | 9310 | 9308 |
| ISAM.PRG | 4782 | 4660 | 4694 | 4846 | 3918 | 3312 | 4258 |
| XLOCK.SBR | 218 | 218 | 218 | 218 | 218 | 218 | 218 |
| FLOCK.SBR | - | - | - | - | 1018 | 1078 | 1078 |
| BASORT.SBR | - | 841 | 877 | 878 | 878 | 878 | 878 |
| AMSORT.PRG | - | 1886 | 1952 | 2018 | 2118 | 2118 | 2118 |
| FLTCNV.PRG | - | 498 | 498 | 498 | 498 | 498 | 498 |

Notes (*) : Monitor Sizes --- MONGEN has problems in Version 4.2; as of now, I am not sure exactly what the size of SYSTEM.MON is really supposed to be, but the above 4.2 sizes were derived from a Persci-AMS system. The corresponding values for a Hawk System were 13494 & 13364. Note that both of these values are different from the Persci System, but that the Unused Monitor Area is still 4314 bytes.
 Driver Sizes --- The 4.2 drivers are created using FIXDVR.

PHILIP A. PUTMAN, & Associates

LAW OFFICES

In reply refer to:

Huntington Beach Executive Park
16168 Beach Blvd. - Suite 141
Huntington Beach, California 92647
(714) 842-4484

August 28, 1979

2121 N. Broadway St.
Santa Ana, California 92706
(714) 547-8338

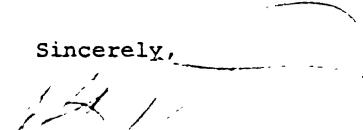
AMUS
P. O. Box 1723
Boulder, Colorado 80306

Gentlemen:

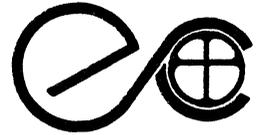
HALP (At's the way we say it back home).
This attorney needs a good applications program
that will do General Ledger, Accounts Payable,
Payroll, Accounts Receivable, Billing with time
rates and trust accounting, Docket Report,
Client/Case Index Report and Employee Billable Time
Reporting.

Contact Phil Putman at (714) 842-4484 and
send documentation to 16168 Beach Boulevard,
Suite 141, Huntington Beach, California 92647 -
with price.

Sincerely,


PHILIP A. PUTMAN

PAP:pk



June 12, 1979

Mr. James Taylor
AMUS
c/o Community Free School
P.O. Box 1724
Boulder, Colorado 80306

Dear Mr. Taylor,

The National Science Foundation has awarded Eastern College a grant for the support of a project entitled Science curriculum development utilizing computer centers based on time-sharing Alpha Microsystems.

The overall goal of the project is to establish three Alpha microcomputer centers and to provide faculty training in computer usage.

One of the major outcomes of this project will be a thorough study of the cost effectiveness and feasibility of basing a computer center around an Alpha microsystem in the small college environment. The project results will assist other colleges in evaluating the use of timesharing microcomputers in the place of more expensive systems. The project includes the implementation of graphics (CRT, Plotters), magnetic tape, as well as using computer-aided-instruction and simulation methods.

We would be very interested in exchange of software and ideas with other users.

Sincerely yours,

A handwritten signature in cursive script that reads "Jonathan C. Barron".

Jonathan C. Barron
Assistant Director, NSF Project

JCB/jll

Steven W. Orr
Parametrics Inc.
221 Crescent St.
Waltham, Ma. 02154

AMUS
c/o Community Free School
P.O. Box 1742
Boulder, Colorado 80306

Dear AMUS,

All of our programs here have no line numbers for purposes of structure. Occasionally though, line numbers can be quite useful for debugging. To that end, I had a simple program called SEQ.BAS which just read in, line by line, the FILE.BAS and produced a FILE.SEQ.

AS of 4.2 however, we now have the ability to format our IF tests by such structures as:

```
IF (CONDITION) &  
    THEN &  
        STATEMENT1 &  
        :STATEMENT2 &  
        :ETC &  
    ELSE &  
        STATEMENT3      ! end of statment
```

The enclosed program will put sequential line numbers onto a .BAS program but not touch the 2nd through last line of a multiple line statement. It may be activated by SEQ.DOC[2,2].

Sincerely,
Steven W. Orr
Parametrics Inc.

SEQ.BAS

```
STRSIZ 136
FALSE=0
TRUE=NOT FALSE
LAST'LINE'DIDN'T=TRUE
INPUT " What is the name of the FILE.BAS to be seq'ed ";FILE$
OPEN #1,FILE$+".BAS",INPUT
OPEN #2,FILE$+".SEQ",OUTPUT
INC=10
INPUT " Resequencing interval ? (default=10) ";INC
LINE'NUMBER=10
INPUT " Starting line number ? (default=10) ";LINE'NUMBER
MORE: INPUT LINE #1,SRC$
      IF EOF(1)=1 &
          THEN &
              GOTO END'IT'ALL
      IF (RIGHT(SRC$,1) # "&" AND LAST'LINE'DIDN'T ) OR &
          (LAST'LINE'DIDN'T) &
          THEN &
              SRC$=STR(LINE'NUMBER)+" "+SRC$ &
                  ;LINE'NUMBER=LINE'NUMBER+INC      ! Up the line #.
      PRINT #2,SRC$
      IF RIGHT(SRC$,1) = "&" &
          THEN &
              LAST'LINE'DIDN'T=FALSE &
          ELSE &
              LAST'LINE'DIDN'T=TRUE
      GOTO MORE
END'IT ALL:
      CLOSE #1
      CLOSE #2
      PRINT "Sequenced file is in ";FILE$+".SEQ"
      END
```

```
SEQ.DOC2,2]
%D $ 10 10
;R
RUN DSK0:SEQ[110,120]
$0
$1
$2
```

4 July 1979

AMUS

c/o Community Free School
P. O. Box 1724
Boulder, Colorado, 80306

Enclosed is a modified version of a program in your May issue by Donald Schaffer. This program addresses the problem of how many blocks of contiguous space are available on the disk. This particular version returns the number of contiguous blocks for the disk the user is logged in to. Any comments would be appreciated.

Thanks

Jeff Oberholtzer

OBERHOLTZER COMPUTER ENGINEERING • 1217 YNEZ AVENUE, REDONDO BEACH, CA 90277 • (213) 378-4096


```

MOV      #OFFSET, R0          ; Extra at end of last BITMAP word
SUB      R0,R2                ; Subtract offset from last record
CMP      R1,R2                ; count and compare again
BGE      LSTCK                ; If R1 < R2,
MOV      R2,R1                ; Move R2 to R1 and
LSTCK:   MOV      BITADR,R2    ; Restore R2
        BIC      #1,@R2        ; Unlock bitmap
EXIT:    LEA      R2,DEV        ; Load storage address
        DCVT     0,10          ; Convert and store record #
        MOV      WHERE,R3      ; Restore arg pointer
        MOV      4(R3),R4      ; Get arg address
        MOV      DEV,(R4)+     ; Return 1st word
        MOV      DEV+2,@R4     ; Return 2nd word
        DELMEM   MDB+2        ; Delete DDB buffer
        RTN
WHERE:   WORD
BITADR:  WORD
DEV:     WORD      0,0
MDB:     BLKW     24
        END

```

June 6, 1979

Mr. Jim Taylor
Alpha Micro Users Society
c/o Community Free School
Box 1723
Boulder, Colorado 80306

Mr. Taylor:

I have an Alpha-Micro System with two CalComp T80 drives. When 4.1 was installed we started experiencing difficulty with COPY (CANNOT DELETE - BITMAP KAPUT, CANNOT DELETE - DEVICE DOES NOT EXIST, CANNOT ACCESS - FILE ALREADY EXISTS). The following changes will clear the problem.

.SET OCTAL
.DDT TRISO.DVR refer to AMOS Operator Instruction Manual for use of DDT
(replace contents of locations 12 and 40 with 57740)
.DDT DSKOPY.PRG
(replace contents of location 742 with 57740)

I am told that there are other Alpha-Micro installations with the CalComp T80 drives but have not been able to contact any. If anyone out there is listening please get in touch with me - an exchange of information would surely be to our mutual benefit.

Sincerely,



Mary Sue Lee, Systems Analyst
Lathrop's Shooters Supplies
5146 E. Pima Street
Tucson, Arizona 85712

Payne, Jackson and Associates
Certified Public Accountants
611 West Ninth Avenue, Anchorage: Alaska 99501
(907) 272-7261 or 279-2351

August 21, 1979

Alpha Micro Users Society:

We have encountered some problems with the CDC Phoenix 90MB disk drive, and have come up with some solutions. I think some of the AMUS members will be interested in reading this. First it is only a 87.2 mega-byte drive (29076 blocks per disk maximum).

Alpha Micro has had a lot of trouble with the PROMs for the AM-410 which is used to boot the system. They sent out all of the first batch of drives with bad PROMs (I've been told). We were told that most of the second batch did not work either. Our's sure didn't work. Well the third PROM we received actually booted the system (more or less).

At least our copy (and possibly all copies) of SMDL0D.PRG (the program on the PROM) has a phantom line problem. If there is memory in the upper 8 or 16k of 64k, it must be disabled in order for the system to boot (there is nothing new about this). Previous XXXL0D programs turned this memory back on using the phantom line when the program was finished. Well, SMDL0D.PRG V4.2.5 does not. The result is that all memory that has the phantom line jumpered will remain in the off condition after the system is up.

To get around this, we have been using memory management to turn on the the top 16k. We simply set the top 16k board in bank 0 to power up in the off position, and do not use the phantom line anywhere in the system. The first time that bank 0 is accessed, memory management will turn on that board. This solution is only possible if there are memory management boards in the system.

The next problem we encountered is that AM changed KILL.PRG in version 4.2.5. Unfortunately, it does not work any more. We are using KILL from version 4.2. It appears as though AM changed KILL in an attempt to solve the following problem.

For as long as I can remember, AMOS has been sensitive to the first character received from jobs. If you turn on your CRT for example after the system has booted, you may lock up your job. If the system is already up when one of us decides to log on, we always unplug the 25 pin cable on the back of the CRT, turn it on, plug it in again, then hit return a couple times. Now, under version 4.2.5 every job is even more sensitive about that first character.

The real problem is in the print spoolers. You can't do the same thing with a print spooler which is a pseudo terminal. We could not use our tried and true SYSTEM.INI to bring up the system. The first job and the first print spooler would hang up and prevent SYSTEM.INI from finishing. We found the solution is to send a blank space " " to the print spooler as the first character after the FORCE line. Our SYSTEM.INI now does the following after the spoolers are killed and the jobmem is set. FORCE SPOOL (return) (blank space) (return) LOG 1,2 (return) LPTINI LPTCEN.INI (return) (return). The process is repeated for each spooler.

After we finally got the system up and running properly, we were faced with the problem of backing up the fixed drives. Since you cannot use the DSKCPY program, you must re-think the problem. (DSKCPY cannot be used due to the bad block problem.) Alpha Micro suggests the use of the COPY program. Well, there is really a lot more to it than simply invoking the COPY program with *.*[*,*]. The BADBLK.SYS prevents it from being a simple copy job. We wrote a basic program which builds a comand file to perform the various steps. It is attached to this letter. The first part of the program is used to help ensure that the operator knows what he/she is doing (we use a five day backup schedule). The program also checks the serial number on the cartridge to ensure that this cartridge is supposed to be used for the backup procedure. It is extremely important that the top fixed disk not have a serial number which is the same as a backup cartrage since the system might boot off the cartridge. Line 2030 has to do with the serial number. PPNFIX on line line 3140 is really SYSACT. We renamed it for security reasons. The last section of the program builds the command file. We have found this to be the fastest way to backup a disk. Even so, it looks like this will take about 1.7 hours when the disk is full.

One last thing, if you are hoping that the Phoenix is so fast that it will speed up your multi-user environment, don't hold your breath. We were operating with Persi drives, and found that with 2, 3 or more users the system degraded. Well so does the Phoenix, and almost as much. The real bottle neck appears to be the computer, not the drive.

I hope that some of the AMUS members can profit from what we have learned about the CDC Phoenix. I would hate to think that everyone has to go through what we went through.


Phillip N. Payne

```

100 MAP1 STRING$,S,512
1000 ?TAB(-1,0);
1010 ?"THIS IS THE PJA PHOENIX BACKUP ROUTINE"
1020 ?"You MUST be running on the fixed disk,"
1030 ?"and have a back up disk loaded for the cartridge."
1050 ?:"Do you wish to continue";
1060 INPUT YESNO$
1070 IF YESNO$$"YES" AND YESNO$$"yes" END
1100 ?"Which disk do you wish to back up";
1110 INPUT INDRIVE
1120 INDRIVE=INT(INDRIVE)
1130 IF INDRIVE=>0 AND INDRIVE<5 GOTO 1200
1140 ?"INVALID DRIVE - DO YOU WISH TO CONTINUE";
1150 YESNO$=" ":INPUT YESNO$
1160 IF YESNO$$"YES" AND YESNO$$"yes" END
1170 GOTO 1100
1200 ?"What day of the week is this";
1210 DAY=0:INPUT DAY$
1220 IF DAY$="MONDAY" OR DAY$="monday" DAY=1
1230 IF DAY$="TUESDAY" OR DAY$="tuesday" DAY=2
1240 IF DAY$="WEDNESDAY" OR DAY$="wednesday" DAY=3
1250 IF DAY$="THURSDAY" OR DAY$="thursday" DAY=4
1260 IF DAY$="FRIDAY" OR DAY$="friday" DAY=5
1270 IF DAY=0 ?"INVALID DAY OF THE WEEK":END
1300 IF INDRIVE=0 AND DAY=1 GOTO 2000
1310 IF INDRIVE=1 AND DAY=2 GOTO 2000
1320 IF INDRIVE=2 AND DAY=3 GOTO 2000
1330 IF INDRIVE=3 AND DAY=4 GOTO 2000
1340 IF INDRIVE=4 AND DAY=5 GOTO 2000
1350 ?"DRIVE & DAY OF THE WEEK DO NOT MATCH":END
2000 OPEN #1,"DSK5:BADBLK.SYSC1,2]",INPUT
2010 INPUT #1,STRING$
2020 CLOSE #1
2030 IF STRING$[1;6]="BACKUP" GOTO 2100
2050 ?"DISK #5 IS NOT A BACK UP CARTRIDGE":END
2100 REM
2200 IF INDRIVE#1 GOTO 2300
2210 LOOKUP "DSK1:BADBLK.SYSC1,2]",RESULT
2220 IF RESULT#1 ?"DISK #1 IS NOT MOUNTED":END
2300 IF INDRIVE#2 GOTO 2400
2310 LOOKUP "DSK2:BADBLK.SYSC1,2]",RESULT
2320 IF RESULT#1 ?"DISK #2 IS NOT MOUNTED":END
2400 IF INDRIVE#3 GOTO 2500
2410 LOOKUP "DSK3:BADBLK.SYSC1,2]",RESULT
2420 IF RESULT#1 ?"DISK #3 IS NOT MOUNTED":END
2500 IF INDRIVE#4 GOTO 2600
2510 LOOKUP "DSK4:BADBLK.SYSC1,2]",RESULT
2520 IF RESULT#1 ?"DISK #4 IS NOT MOUNTED":END
2600 REM
3000 OPEN #1,"DSK0:BAKUP.CMDC1,4]",OUTPUT
3110 ?#1,"":T"
3120 ?#1,"LOG DSK5:1,2"
3130 ?#1,"COPY DSK0:BADBLK.BAK=BADBLK.SYS"
3140 ?#1,"PPNFIX DSK5:"
3150 ?#1,"I"
3160 ?#1,"Y"
3170 ?#1,"A1,2"
3180 ?#1
3190 ?#1,"E"
3200 ?#1,"COPY BADBLK.SYS=DSK0:BADBLK.BAK"

```

```
3210 ?#1,"DSKANA DSK5:/L"  
3300 IF INDRIVE=0 ?#1,"COPY *.*[*,*]=DSK0:.*[*,*]"  
3310 IF INDRIVE=1 ?#1,"COPY *.*[*,*]=DSK1:.*[*,*]"  
3320 IF INDRIVE=2 ?#1,"COPY *.*[*,*]=DSK2:.*[*,*]"  
3330 IF INDRIVE=3 ?#1,"COPY *.*[*,*]=DSK3:.*[*,*]"  
3340 IF INDRIVE=4 ?#1,"COPY *.*[*,*]=DSK4:.*[*,*]"  
3400 ?#1,"COPY BADBLK.SYS=DSK0:BADBLK.BAK"  
3410 ?#1,"SYSTAT"  
3420 ?#1,"LOG DSK0:1,4"  
3450 ?#1,"ERASE DSK0:BAKUP.CMDC[1,4]"  
3500 CLOSE #1  
3600 CHAIN "DSK0:BAKUP.CMD"
```

SOFTWARE AVAILABLE

ACCOUNTING

Jorge Coto, Budget Computer Systems, Miami, FL 305-446-9624

Richard Brock, RB Graphic Supply Co, Garden Grove, CA 714-897-0341.

Henry Arjad, Orange County Computer Center, Costa Mesa, CA
714-646-0221

Don Robertson, Robertson Accounting, Fullerton, CA 714-738-6434

Scott Brim, Computer Systems for Business, Las Vegas, NV
702-873-7400

ANIMAL

Don Jarvis, Electronics Speciality, Houston, TX 713-665-0477
(Pest Control)

AUTO

Terry Woodward, The Computer Center, Birmingham, AL 205-942-8567

BUSINESS

SEE ACCOUNTING

CONSTRUCTION

Don Jarvis, Electronics Speciality, Houston, TX 713-665-0477

Peter Murphy, Cincinnati, OH 513-751-6893

DATA BASE MANAGEMENT

Khalsa Computer Systems, Pasadena, CA 213-684-3311

Dravac, New York, NY 212-889-1556

Byte Shop of Lawndale, Lawndale, CA 213-571-2421

DENTAL

DeMarco Shatz, Torrance, CA 213-533-5080

EDUCATION (SEE ALSO SCHOOLS)

Steve Stoner, Synsys Inc. Des Moines, IA 515-279-8861

GRAPHIC

Peter Jacobson, Absolute Sound, Birmingham, Mich., 313-642-4707

Mark Antin, M. Kerber Inc., New York, NY 212-226-4222

HOSPITAL

Lee Stone, Cmptr C, Topeka, KS 913-233-4433

INSURANCE

Jim Taylor, Data Processing Consultants, Boulder, CO 303-449-8847

INVENTORY

Richard Brock, SEE ACCOUNTING

Jorge Coto, SEE ACCOUNTING

Scott Brim, SEE ACCOUNTING

Don Schaffer, Celestial Seasonings, Boulder, Colorado 303-449-3779

LOGGING

Rod Brewer, Pine Top Logging, Whiteriver, AZ 602-338-4355

MAILING

Phil Theis, Alpha Computer Center, Lancaster, PA 717-229-0567

Peter Burke, Basic Business Software, Salt Lake City, UT
801-565-1199

MEDICAL

Scot Brim, SEE ACCOUNTING

Peter Jax, Automated Professional Systems, Garden Grove, CA
714-891-5516

David Garver, Champaign, IL 217-536-4131

MUSEUMS

Patrick LaFollette, Malacology Section, L.A. Museum of Natural
History, Los Angeles, CA 213-744 5577

PHARMACY

Terry Woodward, SEE AUTO

Turner, Anne, Computer Associates, Inc., Huntsville, AL
205-533-5150

PRINT SHOPS

Richard Brock, SEE ACCOUNTING

PRODUCTION

Don Schaffer, SEE INVENTORY

RESTURANTS

Bob Bruce, Management Data Services, Baltimore, MD 301-857-2250

Business Information Systems, 402-539-8786

RETAIL OPERATIONS

Jorge Coto, SEE ACCOUNTING

Peter Jacobson, SEE GRAPHICS

SCHOOLS (SEE ALSO EDUCATION)

Phil Theis, SEE MAILING

Henry Trist, K-D Enterprises, Sunvalley, CA 213-983-0080

TAXES

Ira Hanson, Mammoth Lakes, CA 714-934-2937

Irvin Eichenbaum, Huntingdon Valley, PA 215-464-1056

TYPESETTING

Lee Stone, SEE HOSPITAL

UTILITIES

Peter Jacobson, SEE GRAPHICS

WORD PROCESSING

Dravac, SEE DATA BASE MANAGEMENT

Khalsa, SEE DATA BASE MANAGEMENT

Matrix Associates, Santa Rosa, CA 707-545-5854

Jim Taylor, SEE DATA BASE MANAGEMENT

Chris Midgley, datalab inc., Ann Arbor, MI 313-995-0663

THIS LIST SHOULD BE UPDATED EACH MONTH.

IF YOU HAVE ANY CHANGES OR ADDITIONS TO THIS LIST, PLEASE
CONTACT SHARON AT P.O. BOX 1723, BOULDER, COLORADO 80306
CR 303-449-6917.

DATA MASTERS

"CUSTOM BUSINESS SERVICES"

4303 SMITH ROAD

CINCINNATI, OHIO 45212

(513) 531-0009

Jim Taylor
Alpha Micro User Society
P.O. Box 1723
Boulder, Colorado 80306

July 19, 1979

Dear Jim:

We are pleased to announce the availability of our Accounts Receivable System. It was written primarily for the use of the health professions; however, works very well as a generalized A/R. This is a balance forward system with unlimited transactions per bill. Some of the main features include family billing, account aging, automatic pricing, automatic description, auto pricing, and description override. The principal outputs are the bills and an aged account list. This system operates on either hard disk or floppy disk systems. Complete documentation is available for \$15. The total system cost is \$300.

We would appreciate it if you would run this in the next AMUS newsletter.

Sincerely,

Roger D. Jeter

Data Processing Consulting, Inc. _____

INSURANCE AGENCY SYSTEM

A complete in-house system for the
Independent Property and Liability Agency

Features: Accounting - More than a service bureau offers.
Marketing - Integrated Word Processing.
Management - Analysis reports, tickler system.

Write or call for a free brochure or send \$10.00 for a 58
page document titled:

EVERYTHING YOU ALWAYS WANTED TO KNOW
ABOUT AIM
BUT WERE AFRAID TO ASK

Also available is a Group Health Administration Billing and
Eligibility system.

DIABLO.DVR - for the 16xx series Diablo printers
.allows Diablo to be run at 1200 baud
.interprets all Diablo functions as TAB(-1,xxx)
. source and documentation available on floppy
for only \$25.00

LETTER.RUN - extention of Alpha Micro Word Processing
If you are using VUE and/or TXTFMT you can
complete your word processing needs with
LETTER for only \$25.00.

Includes: Text insertion
Multi file concatenation
Single page printing

And more
P.O. Box 1723

303 449-8847

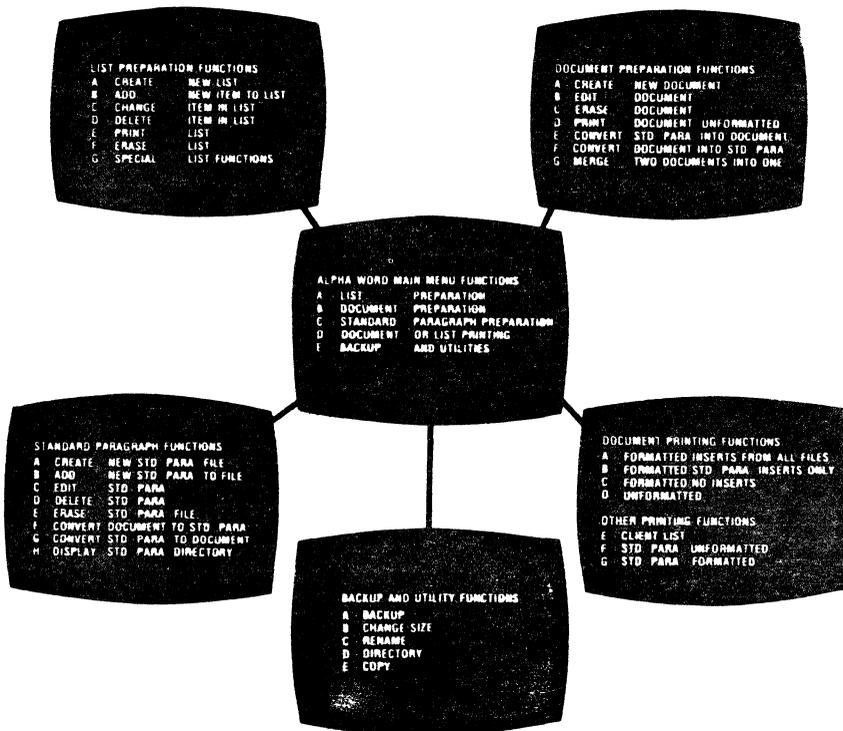
Boulder, Colorado 80306

For those of you with special jobs to do that you cannot
get done yourself, we can provide expertise in ASM
and BASIC. Drivers, interfaces, transaction processing
or any special coding needs. Contact Andy Linch at
D.P.C.I., 1898 S. Flatiron Ct., Boulder, CO 80301

Leasing monies are available for hardware and software such
as turnkey systems on the Alpha Micro. Leasing is
available for dealers and users alike. Governmental
agencies are particularly being sought. Contact Jim
Taylor at D.P.C.I., 1898 S. Flatiron Ct. Boulder, CO 80301

ALPHA WORD™

ALPHA MICRO MENU DRIVEN WORD PROCESSOR



ALPHA WORD is a complete menu driven word processing system written for the Alpha Micro AM-100 computer

The Document and Standard Paragraph processors have the main features of Alpha Micro's VUE text processor, including line insert and delete, word insert and delete, page skipping, global search and replace, automatic linefeed at the end of the line, and block move copy and delete. Your text is interactively stored on disk to ensure system reliability.

The List processor design is a general linked list with user defined data items and data item keys.

The users manual for ALPHA WORD is written in such a manner that your secretary can easily learn the system in a few hours. This is possible through the use of extensive examples and an easy to read style.

Enhancements appearing in future revisions of ALPHA WORD will be made available to registered users for \$30 on floppy disk. Future versions will include (1) Spelling error correction and detection, (2) Automatic index and table of contents generation, (3) Half justification and automatic hyphenations, (4) List search display similar to IBM System 6.

ALPHA WORD software is shipped on an 8 inch floppy disk in standard or AMS format, or on a hard disk for an additional \$100. Demonstration disks, including users manual, are available for \$35 or users manual separately for \$15 each.

ALPHA WORD software has been reduced by popular demand to

\$600

with quantity discounts to dealers.



317 West Broadway
Suite 11
Eugene, Oregon 97401
(503) 484-1040

Utilities Package #2 for the Alpha Micro

The Real Oregon Computer Company is announcing the availability of the second in a series of Alpha Programmer's Utilities Packages. The second package consists of nine assembly language subroutines, each with source code and documentation.

The package includes a very valuable subroutine which reports on the maximum number of CONTIGUOUS free blocks on a disk. Also included are five subroutines to make menus, submenus, or operator's instructions, a very easy and fast task to program. Another subroutine allows the program to check the input buffer. This routine can be used to stop processing or direct printing by simply pressing any key on the keyboard.

The package is \$35.00 distributed on floppy disk and \$125.00 distributed on a Hawk cartridge. To take advantage of these utilities, order yours today from The Real Oregon Computer Company, 317 West Broadway, Suite 11, Eugene, Oregon, 97401; or phone your order to (503) 484-1040.

Alpha Programmers Utility Package #1

To make your programming effort more productive, here is a list of ROCC/Alpha Micro subroutines, Basic and assembly language programs on AMS/IBM diskettes and HAWK cartridges.

UTILITIES

- RENUMB - Renumbers and denumbers lines in files (e.g. BASIC programs). Ability to add line numbers to a file entered without them. Extremely fast and efficient. Well worth the price of the package by itself.
- XREF - Generates cross reference listing of variables, labels, reserved words used in BASIC programs. Processes about 10 blocks of source code per minute. Very fast.
- RANCPY - Copies "N" blocks of a random file to a second file. Very useful for Hard disk to Floppy disk transfer of large files, resizing files, partial file copies, etc.
- HEAD - Enables characters typed on the CRT to be printed on the printer. Useful if your printer has no keyboard.
- SAVUE - Saves MEM:IMAGE.VUE files on disk, putting in the appropriate LF and CR.

SUBROUTINES

- XFERBY - Designed to do byte transfers of string variables; to get around Alpha Basic's habit of padding spaces to the left of an unformatted string transfer.
- XSHELL - An efficient and flexible subroutine to sort arrays in Basic programs
- XSCRAN - Scans a string for symbols. This is used in the XREF program.
- XTREES - Binary tree search for the XREF program.

All subroutines are clearly documented. Pricing: \$35 for the diskette version and \$125 for the HAWK cartridge version. The price includes all source and object programs plus documentation files for each module. The RENUMB program is enough to cost justify the investment just in time saved typing line numbers. Dealer inquiries invited. Order yours today from:

The Real Oregon Computer Company
317 West Broadway, Suite 11
Eugene, Oregon 97401

Computer Systems and Services for Business



317 West Broadway
Suite 11
Eugene, Oregon 97401
(503) 484-1040

Alpha Programmers Utility Package #2

To make your programming effort more productive, here is a list of ROCC/Alpha Micro Basic Xcall subroutines and utilities on AMS/IBM diskettes and HAWK cartridges.

XCALLS

XMENU, menu\$, delimiter\$, y, x
Displays a menu on the screen..

ASKOPT, options\$, index, immediate'mode
Asks which option the user wishes to choose.
Will accept only valid options.

XCLR, ytop, xleft, ybottom, xright
Clears a rectangular area on the screen.

XBOX, ytop, xleft, ybottom, xright [,char (s)]
Draws a box on the screen.

XGRAB, number'of'characters, characters\$
Accepts characters from input buffer without the use of "input" statement. Useful to control printing or processing.

XFREE, device'name\$, free'space, max'contig'area
Reports the number of free blocks and maximum number of contiguous free blocks on the specified device.

XDOWN, string\$
Displays a character string vertically on the screen.

XDATE, mm-dd-y,yyymmdd
Returns the system date in two formats (mm-dd-yy and yyymmdd).

XLOOK, program'name'in'memory, found
Looks for a program or subroutine in memory.

All subroutines are clearly documented. Pricing: \$35 for the diskette version and \$125 for the HAWK cartridge version. The price includes all source and object programs plus documentation files for each module.

The Real Oregon Computer Company
317 West Broadway, Suite 11
Eugene, Oregon 97401

CROSSROADS CENTER
 4104 S. KIETZKE LANE
 RENO, NV 89502
 (702) 826-8080

June 25, 1979

Jim Taylor
 AMUS
 c/o Community Free School
 PO Box 1724
 Boulder, CO 80306

Dear Jim,

Enclosed is the source listing of an assembly language program which converts numbers between floating point and binary. The routine is easy to use, requiring the calling program to just push four words onto the stack, and then call FP.PRC (See the program listing).

I have found that the easiest way to use this program is to store FP.PRG (My name for this program) as a memory module and utilize the SRCH macro to locate FP.PRG and save the address somewhere. This way, it can be used as an external subroutine, and doesn't have to be inserted into the user's program.

Here is a simple program which demonstrates FP:

```
;Convert the floating point number in F to binary in B
COPY      SYS
LEA       RC,F           ; push address floating point
PUSH     RC
LEA       R0,B          ; push address of binary number
PUSH     R0
PUSH     #2             ; size of binary in bytes
PUSH     ; function=0: truncate
SRCH     FPNAME,R0     ; search for program in memory
BNE      FPNF           ; couldn't find it
CALL     @R0            ; call FP
MOV      B,R1          ; display result
DCVT     0,2
CRLF
EXIT

FPNF:    TYPECR ?FP.PRG not in memory
EXIT

B:       BLKB          2           ; binary
F:       WORD          41300,0,0  ; floating point constant 24.0
FPNAME:  RAD50        /FP   PRG/  ; nameblock
END
```

I hope that other assembly language programmers will find this routine useful in their own applications.

Sincerely,



John Waycott

```

;*****
;*
;*      .      floating point conversion routine
;*
;*****

```

```

;Author: John Waycott, The Byte Shop of Reno
;

```

```

;Calling sequence:

```

```

;   PUSH    floating point variable address
;   PUSH    binary variable address
;   PUSH    binary variable size (in bytes)
;   PUSH    function: 0=Truncate - convert floating point to binary
;               2=Float   - convert binary to floating point
;   CALL    FP.PRG

```

```

FP:   COPY    SYS
      SAVE
      MOV     #200,R4      ; save registers
      TJMP   16(SP)      ; constant used throughout
      OFFSET TRUNC       ; jump to desired function
      OFFSET FLOAT

```

```

;*****
;*      trunc      *
;*****

```

```

;convert real to binary

```

```

TRUNC: MOV     24(SP),R5      ; address float point var
      MOV     (R5)+,R2      ; and put it in r2:r1:r0
      MOV     (R5)+,R1
      MOV     @R5,R0
      CLR     -(SP)        ; clear sign flag
      TST    R2            ; if number is negative,
      BPL    .+10         ; set sign flag and
      BIC    #10000,R2    ; take the absolute value
      SET    @SP
      MOV    R2,R3        ; put exponent in r3
      SSRA  R3,7
      SUB    R4,R3
      BLE    TZERO       ; number is less than 1.0
      BIC    #177400,R2  ; clear exponent from number
      BIS    R4,R2       ; set implied bit
      SUBI   10,R3       ; determine word size
      FLE    T256       ; 1 byte (n<256)
      CMP    R3,#10     ; is number <65536?
      BLE    T65536     ; yes
      SUB    #40,R3     ; shift r2:r1:r0 right 40-exp times
      BGT    TZERO      ; overflow
      BEQ    TCEND
      NEG    R3
      LCC   0
      RORB  R2
      ROR   R1
      ROR   R0
      SOB  R3,.-10

```

```

        BR      TCEND
;truncate r<256
TZ56:   ASH     R2,R3          ; shift the number right
        BR      TX5B          ; translate to 5-bytes
;truncate r<65536
TZ5536: ASHC    R1,P3          ; shift r2:r1 to left
TX5B:   MOV     R2,R0          ; translate to 5-bytes
        CLR    R1
        CLR    R2
;5-byte binary is in r2:r1:r0
TCEND:  TST     (SP)+          ; was sign bit set?
        BEQ    STRSLT         ; no
        COMB   R2             ; yes - negate result
        COM    R1
        COM    R0
        INC    R0
        ADC    R1
        ADC    R2
;store binary r2:r1:r0 into user's variable
STRSLT: MOV     20(SP),R5      ; size of binary
        MOV     22(SP),R4      ; address of binary
        MOVEB   R0,(R4)+
        DEC     R5
        BEQ     FPEXIT         ; 1 byte
        SWAB   R0
        MOVB   R0,(R4)+
        DEC     R5
        BEQ     FPEXIT         ; 2 bytes
        MOVB   R1,(R4)+
        DEC     R5
        BEQ     FPEXIT         ; 3 bytes
        SWAB   R1
        MOVB   R1,(R4)+
        DEC     R5
        BEQ     FPEXIT         ; 4 bytes
        MOVEB   R2,(R4)+
        DEC     R5
        BEQ     FPEXIT         ; 5 bytes
        BR      FPEXIT
TZZERO: CLR     R0             ; return binary zero
        CLR    R1
        CLRB   R2
        BR     TCEND

;*****
;*      float      *
;*****
;convert binary to real
FLOAT:  MOV     20(SP),R0      ; size of binary
        MOV     22(SP),R5      ; address of binary
        CLR     -(SP)          ; clear 6 bytes on stack
        CLR     -(SP)
        CLR     -(SP)
        MOV     SP,R1          ; move binary to stack
        MBBU   R5,R1
        POP    R0             ; load binary into r2:r1:r0
        POP    R1

```

```

      POP      R2
      CLR      -(SP)          ; clear sign
      TSTB     R2             ; is number zero?
      BEQ      TST2          ; first word is zero
      EPL      FLTFUL        ; its positive
      SET      @SP           ; negative - set sign and complement
      COMB     R2
      COM      R1
      COM      R0
      INC      R0
      ADC      R1
      ADC      R2
TST2:  TST      R1             ; is next word zero?
      BNE      FLTFUL        ; no - full conversion
      TST      R0
      BEQ      FZERO         ; entire number is zero
;The number is less than 65536
      CMP      R0,#256.       ; is it <256?
      BHIS     F65536        ; no, but its one word
;convert number<256 to floating
      MCV      #42200,R2     ; exponent
      SUB      R4,R2         ; shift out implied bit
      ASLB     R0
      BCC      .-4
      LCC      0
      RORB     R0
      BISB     R0,R2         ; set real in r2:r1:r0
      CLR      R1
      CLR      R0
      BR       FLTEND
;convert b<65536 to real
F65536: MOV      #44000,R2    ; set exponent
      MCV      R0,R1         ; put number in r1
      PMI      .+10         ; no need to shift
      SUB      R4,R2         ; shift out implied bit
      ASL      R1
      EPL      .-4
      SWAB     R1           ; shift over 9 bits
      BIC      R4,R1
      BISB     R1,R2
      CLRB     R1
      CLR      R0
      BR       FLTEND
;Full 5-byte conversion
FLTFUL: MOV      #52000,R3    ; set exponent
FSH:   SUB      R4,R3        ; decrement exponent
      ASL      R0
      ROL      R1
      ROLB     R2
      EPL      FSH          ; continue until msb is set
      BIC      #177600,R2    ; discard msb (implied bit)
      BIS      R3,R2         ; set exponent
FLTEND: TST      (SP)+       ; sign bit set?
      BEQ      .+6          ; no
      BIS      #100000,F2    ; yes - set sign bit

```

```
      MOV      24(SP),R5          ; address floating var
      MOV      R2,(R5)+         ; return floating result
      MOV      R1,(R5)+
      MOV      R0,@R5
EPEXIT: REST                    ; restore registers
      MOV      @SP,10(SP)       ; set up for return
      ADDI     10,SP           ; discard stuff on stack
      RTN
FZERO: CLR     R0              ; return zero
      CLR     R1
      CLR     R2
      BR      FLTEND
      END
```

ALPHA MICRO USERS SOCIETY Membership form

Please fill out as much information as possible.

Name _____ Company _____

Address _____ City _____ State _____ Zip _____

Business Phone _____ Home phone _____

Circle one: Own Lease Thinking

Check all applicable: Dealer OEM User: Corporate Individual

Describe equipment: _____

AMUS may use my name for mailing lists Make checks payable to AMUS

Annual dues are \$25.00 per member.

AMUS
c/o Community Free School
P O Box 1724
Boulder, Colorado 80306

NON-PROFIT ORG.
U.S. POSTAGE
PAID
Permit No.490
Boulder, Co. 80302